In Defense of Knowledge Distillation for Task Incremental Learning and its Application in 3D Object Detection

Peng YUN1 Yuxuan LIU1 and Ming LIU1

Abstract—Making robots learn skills incrementally is an efficient way to design real intelligent agents. To achieve this, researchers adopt knowledge distillation to transfer old-task knowledge from old models to new ones. However, when the length of the task sequence increases, the effectiveness of knowledge distillation to prevent models from forgetting old-task knowledge degrades, which we call the long-sequence effectiveness degradation (LED) problem. In this paper, we analyze the LED problem in the task-incremental-learning setting, and attribute it to the inevitable data distribution differences among tasks. To address this problem, we propose to correct the knowledge distillation for task incremental learning with a Bayesian approach. It additionally maximizes the posterior probability related to the data distributions of all seen tasks. To demonstrate its effectiveness, we further apply our proposed corrected knowledge distillation to 3D object detection. The comparison between the results of increment-at-once and increment-in-sequence experiments shows that our proposed method solves the LED problem. Besides, it reaches the upper-bound performance in the task-incremental-learning experiments on the KITTI dataset. The code and supplementary materials are available at https://sites.google.com/view/c-kd/.

Index Terms—Probability and Statistical Methods; Incremental Learning; Computer Vision for Transportation.

I. INTRODUCTION

Learning skills on top of previous knowledge is a marked feature of intelligent agents. When human beings acquire new knowledge, old skills get preserved, which intensifies our adapting ability to survive in the changing world. Robots run in the same changing world where the data distributions exhibit the long-tail property (Figure 1(a)) [1]. The massive edge-case data points, lying at the tail of the data distribution, are always unexpected in advance. Take autonomous driving as an instance. Researchers have access to common data in driving scenes through public datasets, but must spend huge effort to enumerate and collect edge-case data points, like different types of trucks and various animals. Many tragic autonomous car accidents can be attributed to unexpected edge cases.

Designing optimal incremental learning algorithms to make robots learn skills incrementally is challenging. It has been recently proved that the optimal incremental learning algorithm is theoretically NP-HARD and requires the perfect memory condition [2]. Researchers have made efforts to prevent parametric models from forgetting old knowledge when learning new tasks, i.e., to overcome catastrophic forgetting. Knowledge distillation [3] is one of the methods to prevent parametric models from the catastrophic forgetting. By adding a regularization penalty, it transfers the knowledge from the old model to the new model recursively. It has shown great success in the image classification [4], [5] and image-based 2D object detection tasks [6]–[8].

Knowledge distillation takes effect in reducing forgetting, however, transferring knowledge in the recursion way accumulates to build-up errors. When the length of the task sequence increases, the effectiveness of knowledge distillation drops significantly (Figure 1(b)), which we call the long-sequence effectiveness degradation (LED) problem. Since the unexpected data haunts and frequently appears, researchers have to improve the models on hand repeatedly. In consequence, the LED problem hinders the practical usage of knowledge distillation in incremental learning.

In this paper, we aim to solve the LED problem of knowledge distillation for task incremental learning (TIL), which...
is the incremental learning scenario to overcome the long-tail distributions in the real world. We first analyze the reason for LED problem and attribute it to the inevitable data distribution differences in TIL. To address this problem, we propose to correct knowledge distillation for TIL with a Bayesian approach. Compared with the original knowledge distillation, our proposed method additionally maximizes the posterior probability related to the data distribution of all seen tasks (Figure 1(c)). Finally, we demonstrate the effectiveness of the proposed corrected knowledge distillation with an application of 3D object detection. We consider two incremental learning settings: incrementing tasks at once and incrementing tasks in sequence. The experimental results demonstrate the manifestation of LED problem and the effectiveness of our proposed method.

II. RELATED WORK

In this section, we first review the regularization-based incremental learning methods and describe the relationship between our proposed method and the existing works, and then review the literature of LiDAR-based 3D object detection.

A. Regularization-based incremental learning

The regularization-based methods alleviate old-task forgetting by adding regularization terms in the objective function. Researchers resolve the catastrophic forgetting with this idea in two general ways: knowledge distillation and reserving priors.

a) Knowledge distillation: Knowledge-distillation-based methods [4]–[8] retain the knowledge of previous tasks by restricting the conditional distribution computed with the updated parameters \( p(\hat{Y}|X, \theta) \) close to that computed with the optimal parameter of previous tasks \( p(\hat{Y}|X, \theta_0) \). A regularization term, proportional to the distance between these two conditional distributions, is added to the original loss function.

In classification, the distance is commonly measured by the Kullback-Leibler (KL) divergence [4], [5]. It can be seamlessly extended from classification to regression tasks by replacing the KL divergence with a smoothed-L1-norm or L2-norm. Shmelkov et al. [6] first proposed an incremental learning object detector, called ILOD, and extended the knowledge distillation to the image-based 2D object detection problem. In recent years, there are follow-up works of ILOD [6], including RILOD [7] and Faster-ILOD [8].

Ramen et al. [5] first reported the shortcomings of the knowledge-distillation-based incremental learning methods, and designed EBLL to deal with the data distribution differences for image classification tasks. They proposed to project features on a low dimensional manifold with an under-complete auto-encoder and impeded the new feature deviating from previous task optimal ones. However, it requires additional computation in auto-encoder training and needs to collect low-dimensional features in each optimization step. In contrast, we deal with the inevitable data distribution differences in TIL from a probability perspective by maximizing the posterior probability related to the data distributions of all seen tasks. Our proposed method does not cause an additional computational burden when learning new tasks.

b) Reserving priors: Kirkpatrick et al. [9] formulated the statistical risk in incremental learning with the posterior probability \( p(\theta|D) \) to find the most probable weights given the data \( D = D_A \cup D_B \), where \( D_A \) and \( D_B \) denote old-task data and new-task data respectively. They optimized the parameters \( \theta \) by maximizing the logarithm posterior probability:

\[
\log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B|D_A),
\]

where the term \( \log p(\theta|D_A) \) was modeled with Laplacian approximation with the mean given by the optimal parameters \( \theta_A \) and the covariance matrix approximated by a diagonal Fisher information matrix (FIM). The FIM was computed and stored right after training the old task and worked as the prior information of the old tasks. There are multiple follow-up research on the computation of the weight importance measurements [10]–[14]. Zenke et al. [10] proposed an online method for estimating the weight importance. Instead of computing the FIM in a separate stage after training, they estimate the importance of each parameter calculated with gradients and parameter changes when training the neural network. Aljundi et al. [11] redefined the importance weights to an unsupervised setting. Instead of calculating the gradients of the loss function, they adopt the gradients of the squared L2-Norm of the network output, which can be considered as a heuristic approximation of the FIM. We adopt the method of evaluating the posterior probability on the seen task data distributions in EWC [9] and also consider its alternative MAS [11] in our experiments.

In recent days, Liu et al. [15] reported that EWC failed in the image-based 2D object detection problem, and proposed IncDet to facilitate the use of EWC in incremental 2D object detection by utilizing pseudo bounding box annotations. Their proposed pseudo-annotation technique exploits the old-task optimal model to generate fake labels offline in order to remedy the lack of old-task class annotations in the new-task data. However, this is a suboptimal solution for the parametric models which require online data augmentation, like randomly translating or rotating objects in LiDAR-based 3D object detection. The incorrect pseudo annotations induce a mass of noise in the training process. We compare our proposed method with IncDet [15] in Section V.

B. LiDAR-based 3D object detection

The geometry information captured by LiDARs can be used for perception, and the accurate spatial information is helpful for precise 3D object detection. The challenge for LiDAR-based 3D object detection is that the point clouds captured from LiDARs suffer from the sparsity and are represented as unordered vector lists, which is not suitable for convolution operations. Researchers who intend to percept key objects from LiDAR scans have to deal with the problem: How to extract features from point clouds of LiDARs? The current solution can be categorized into two general categories: Quantization+Convolution NeuralNetwork (Q+CNN) based methods, which convert the input point clouds into convolution-friendly representations and then apply CNNs to extract features [16]–[23], and PointNet-based methods, which
directly extract features from the input point clouds based on PointNet [24] and its variants [25]–[27].

In this paper, we focus on the 3D region proposal networks (RPNs) within the Q+CNN-based methods. The quantization process, which converting input point clouds into convolution-friendly representations, include projecting the point cloud into bird’s-eye-view images [16]–[19] or voxelizing the point cloud into a grid [20]–[23]. In quantization, the 3D spatial information is encoded into bird’s-eye-view images or grids with hand-crafted features, like point density, distance, occupancy, etc. Li et al. [20] encoded the 3D spatial information of point clouds into grids by hand-crafted features and proposed an encoder-decoder network with 3D dense convolutions by extending a 2D fully convolutional network. Zhou et al. [21] proposed an end-to-end network for 3D object detection, called VoxelNet, where the voxel-wise features are learned from raw point clouds instead of hand-crafted by researchers. After quantization, high-level features can be extracted with CNNs and further used for 3D bounding box estimation.

The RPN was first proposed in [28] for image-based 2D object detection task. In [28], an RPN takes an image as input and outputs a set of rectangular object proposals, each with an objectness score. The similar idea of RPN is extended to LiDAR-based 3D object detection tasks in [18]–[22]. In 3D object detection tasks, the RPN finally estimates a classification map and a regression map based on a predefined anchor map. In specific terms, the anchor map contains a set of anchors locating on each discrete location in the 3D space. The classification map predicts the semantic class of each anchor, and the regression map estimates the refined shape and location of 3D bounding box for each anchor.

In recent years, Yan et al. [22] proposed SECOND based on VoxelNet [21] and adopted spatially sparse convolution to deal with the sparsity of point clouds. As a result, the running-time performance of SECOND was improved by 4× (230 ms to 50 ms per frame). We adopt SECOND as the basic 3D object detector for the use case in 3D object detection. For more details about the the network architecture, please refer to the original paper [22].

III. METHODOLOGY

In this section, we first provide the formal definition of TIL and then analyze the reason for the LED problem. Based on the analysis, we further describe our Bayesian approach to correct knowledge distillation for TIL.

A. Problem definition of TIL

We define the TIL following the definition in [29] and consider training a parametric model on a sequence of tasks. Each task task-t consists of a task-specific class set C(t) and a task-specific data distribution (X(t), Y(t)) ∼ P(t). Different tasks have different class sets and data distributions, i.e., C(i) ≠ C(j) and P(i) ≠ P(j), if i ≠ j. The goal of TIL is to control the statistical risk of all seen tasks given limited or no access to data (X(t), Y(t)) from previous tasks t ≤ T:

\[
\sum_{t=1}^{T} E_{(X(t), Y(t)) \sim P(t)} [\ell(f_t(X(t), \theta), Y(t))],
\]

where \( f_t(\cdot, \theta) \) is the parametric model of task t, \( \ell \) is the loss function, and \( T \) is the number of tasks seen so far. For the current task task-T, the statistical risk can be approximated by the empirical risk:

\[
\mathcal{L}(\theta, D^T) = \frac{1}{N_T} \sum_{i=1}^{N_T} \ell(f_t(X_i^T, \theta), Y_i^T),
\]

where the dataset of task-t is sampled from its task-specific data distribution, \( \{(X_i^T, Y_i^T)\} = D^T \sim P(i) \), and \( N_T \) is the capacity of \( D^T \).

The major challenge of TIL is that the data is no longer available for previous tasks when training on new tasks, which hinders the evaluation of statistical risk for the new parameter values on previous tasks.

B. Analysis of the LED problem

Knowledge-distillation-based methods prevent parametric models from forgetting old-task knowledge by restricting the conditional distribution \( p(\tilde{Y}|X, \theta) \) close to \( p(Y|X, \theta^*_1..T-1) \) with the objective function:

\[
\mathcal{L}_{KD}(\theta, \theta^*_1..T-1, D^T) = \mathcal{L}(\theta, D^T) + \lambda \sum_{i=1}^{T-1} KL(p(\tilde{Y}|X_i^{T-1}, \theta) || p(Y|X_i^{T-1}, \theta^*_1..T-1)),
\]

where task-T is the current task, the \( \theta^*_1..T-1 \) is the optimal parameters of the old tasks, and the KL denotes the Kullback-Leibler divergence, \( \lambda \) is the weight factor of the regularization term.

To find the reason for the LED problem, we first consider a two-task case where the task sequence contains task-A and task-B. The statistical risk of TIL is

\[
\mathcal{L}(\theta, D^A) + \mathcal{L}(\theta, D^B)
= E_{(X_A,Y_A) \sim P_A} [\ell(f(X_A, \theta), Y_A)]
+ E_{(X_B,Y_B) \sim P_B} [\ell(f(X_B, \theta), Y_B)].
\]

The objective function of knowledge distillation is

\[
\mathcal{L}_{KD}(\theta, \theta^*_A, D^B)
= E_{X_B \sim P_B} KL(p(\tilde{Y}|X_i^{B}, \theta) || p(Y|X_i^{B}, \theta^*_A))
+ E_{(X_B,Y_B) \sim P_B} [\ell(f(X_B, \theta), Y_B)],
\]

where we drop the hyperparameter \( \lambda \) for a better comparison. It requires two conditions to make optimizing equation (6) equivalent to optimizing equation (5): (1) \( \theta^*_A \) is optimal for \( \ell \) conditioned on the data distribution \( P^A \); (2) the data distribution \( P^A \) is highly related to \( P^B \). In practice, when incrementally training a parametric model on the task-B, we always start with the parameters having converged on the task-A, which is good enough for \( \ell \) conditioned on \( P^A \).

The second condition does not hold in TIL. According to the problem definition of TIL, different tasks have different data distribution \( P(i) \neq P(j) \), and the data distributions can be significantly different among tasks in practice. The tasks can be composed of data points sampled from different generative distributions, as shown in Figure 2 (left). Moreover, data points
of different tasks can also be generated from a joint generative distribution but with different sampling strategies. This is the situation where the parametric model progressively learns the tail classes on a given dataset. To illustrate this point, we simulate this situation based on the KITTI 3D detection dataset [31], and plot the histograms of each separate tasks (task-1 to task-5) and that of the whole dataset (task-1,.5) in Figure 2 (right).

The inevitable data distribution differences in TIL make optimizing the knowledge-distillation loss not equivalent to optimizing the statistic risk for all seen tasks. It results in degradation of old-task performance when adding a new task, and the degradation is relative to the extend of the data distribution differences between the new task and the previously seen tasks. Indeed, it has been shown empirically in [5], [32] that the use of significantly different data distributions leads to a significant decrease in performance for the knowledge-distillation-based method LwF [4].

The degradation will be accumulated to build-up errors, so that it will eventually cause the LED problem when the length of the task sequence increases. In Figure 2 (a), the intersection of the low error regions of task-A and task-B denotes the low error region of TIL. It overlaps with the low error region of $L_{KD}$ (white circle), but they do not completely coincide. When applying knowledge distillation to the third task, the low error region of $L_{KD}$ will drift towards the task-C (Figure 3 (b)). In practice, it can be observed that the parametric model tends to over-fit the new task and the performance on task-A and task-B degrades.

C. A Bayesian solution to correct knowledge distillation

Since the data distribution differences among tasks lead to the LED problem in TIL, we intend to correct the objective function of knowledge-distillation (equation (4)) by adding a constraint related to the data distributions of all previously seen tasks. It can be achieved by maximizing the logarithm posterior probability $\log p(\theta| \cup_{t=1}^{T-1} D^{(t)})$. Therefore, the corrected objective function of knowledge distillation is

$$\theta^* = \arg\min_\theta \mathcal{L}_{KD}(\theta, \theta_1^{*,T-1}, D^T) - \beta \log p(\theta| \cup_{t=1}^{T-1} D^{(t)}),$$

(7)

where $\beta$ is the weighting factor of the logarithm posterior term.

The evaluation of the term $\log p(\theta| \cup_{t=1}^{T-1} D^{(t)})$ is challenging, since the data of previous tasks is intractable when training the task-$T$. Here we adopt the Laplacian approximation in EWC [9] to evaluate this term. Its mechanism is to restore the prior information of old tasks, like FIM or its alternatives, and then use the priors to evaluate this logarithm posterior term for correcting the drift towards the new task. It can be written as

$$- \log p(\theta| \cup_{t=1}^{T-1} D^{(t)}) \approx \frac{1}{2} \sum_{t=1}^{T-1} \sum_i \beta_i F^i_1 [\theta_i - \theta_i^{*,T-1,i}]^2,$$

$$F^i = \frac{1}{|S|} \sum_{D_t \sim D_i} \left[ \frac{\partial}{\partial \theta} \mathcal{L}(\theta, D^t) \right] \frac{\partial}{\partial \theta} \mathcal{L}(\theta, D^t)],$$

(8)

where $\beta_i$ is the hyperparameter balancing the weight of each task-$t$. $F^i_1$ is value of the $i$-th parameter in the diagonal of the FIM computed on task-$t$, and $|S|$ denotes the number of times $D_i$ is sampled from $D_i$. We provide an integrated derivation in our Supplementary Materials.

As a result, the corrected objective function of knowledge distillation for TIL is

$$\mathcal{L}_{C-KD} = \mathcal{L}_{KD} + \frac{1}{2} \sum_{t=1}^{T-1} \sum_i \beta_i F^i_1 [\theta_i - \theta_i^{*,T-1,i}]^2.$$

(9)

Figure 3 (c) illustrates the effects of the corrected objective function. Optimizing the corrected objective function $\mathcal{L}_{C-KD}$ will lead the parameters to the direction maximizing posterior distribution related to all seen tasks, and prevent the optimization process from over-fitting the new task.
Fig. 4: Overview of the dual network learning framework. RPN_A is the well-trained optimal parametric model $f_A(\cdot; \theta_A^*)$ for previous tasks, the parameters of which are frozen. RPN_B is the new-task parametric model $f_B(\cdot; \theta_B)$, the parameters of which are initialized from $\theta_A^*$. The blue arrows represent the backward propagation paths of optimization.

IV. AN APPLICATION IN 3D OBJECT DETECTION

In this section, we demonstrate a use case of our corrected knowledge distillation by applying it to a 3D object detection Region Proposal Network (RPN) under the TIL setting. We have reviewed the background of 3D object detection RPNs in Section II, and now we will state it in a more formal way. A 3D object detection RPN is a parametric model $f_{\theta} = h_{\phi} \circ g_{\phi}$, where $g_{\phi}$ represents the feature extraction submodel, projecting the input into the embedding space $\mathbb{R}^F$, where $F$ is the dimension of the embedding space, and $h_{\phi}$ represents the header of RPN. The $h_{\phi}$ consists of two parts: $h_{\phi,cls}: \mathbb{R}^F \rightarrow \mathbb{R}^{M \times A \times C}$ and $h_{\phi,reg}: \mathbb{R}^F \rightarrow \mathbb{R}^{M \times A \times S}$ conditioned on the anchor map $\alpha \in \mathbb{R}^{M \times A \times S}$ where $M$ is the total number of locations, $A$ is number of anchors on each location, $C$ is the total number of classes, and $S$ is the dimension of the parameterized anchor vector $\mathbf{v}$. For simplicity, we denote $f_{cls} = h_{\phi,cls} \circ g_{\phi}$ and $f_{reg} = h_{\phi,reg} \circ g_{\phi}$ in the rest of the paper.

In 3D object detection, we write the corrected objective function (equation (9)) into the following form:

$$L_{C-KD} = L_{\text{det}}(\theta, D^T) + L_{\text{dist}}(\theta, \theta_{A,T-1}^*, D^T) + L_{\text{MAP}}(\theta, F^1, \ldots, F^{T-1}, \theta_{1,T-1}^*, D^T), \tag{10}$$

where $L_{\text{det}}(\theta; D^T)$ denotes the likelihood of $D^T$ conditional on $\theta$, $L_{\text{dist}}(\theta, \theta_{A,T-1}^*, D^T)$ denotes the knowledge-distillation regularizer, and the last term denotes the logarithm posterior term.

To evaluate the three terms in equation (10), we adopt the dual network learning framework in [6]. We consider the step of incrementally training the task-$T$, $T > 1$, and denote the well-trained optimal parametric model for the previous tasks task-$t$, $t < T$ as $f_A(\cdot; \theta_A^*)$, while represent the current task-$T$ as task-B and its parametric model as $f_B(\cdot; \theta_B)$. Figure 4 demonstrates the dual network learning framework. We forward point clouds $\chi^T$ to $f_A(\cdot; \theta_A^*)$, which is the optimal model for task-$t$, $t < T$. We sample a foreground subset $\tilde{y}_c^A, \tilde{y}_r^A$ from the $RPN_A$ outputs $y_c^A, y_r^A$ with a biased sampling strategy $y_c^A = f_A(\cdot; \chi^T, \theta_A^*)$, $y_r^A = f_A(\cdot; \chi^T, \theta_A^*)$, and then find their corresponding estimations $\hat{y}_c^A, \hat{y}_r^B$ from the $RPN_B$ outputs. As in [LOD] [6], we compute the knowledge-distillation regularization term $L_{\text{dist}}$ with

$$L_{\text{dist}} = \ell_c(\hat{y}_c^A, \tilde{y}_c^B) + \alpha \ell_r(\hat{y}_r^A, \tilde{y}_r^B), \tag{11}$$

where $\ell_c$ and $\ell_r$ represent the distance measurement function for classification and bounding box regression, and $\alpha$ balances the weights of these two terms. The logarithm posterior term is evaluated with $\theta_{A,T}^*$ and $\theta_B$ as well as the data prior FIMs:

$$L_{\text{MAP}} = \frac{1}{2} \sum_{t=1}^{T-1} \sum_i \beta_i \|\theta_{A,i} - \theta_{B,i}\|^2_2. \tag{12}$$

We can compute the FIM with equation (8) in a supervised method. There is also a heuristic computation method MAS [11], which provides an unsupervised way to approximate the FIM with

$$F^t = \frac{1}{|S|} \sum_{\tilde{D} \sim D^t} \left( \frac{\partial}{\partial \theta} \|f_{1..s}(\chi^T, \theta_{1..s}^*)\|^2 \right)_T \times \frac{\partial}{\partial \theta} \|f_{1..s}(\chi^T, \theta_{1..s}^*)\|^2, \tag{13}$$

where the notation is the same as before. We consider both of these two computation methods in our experiments.

V. EXPERIMENTS

In this section, we report the experimental results to demonstrate the effectiveness of our proposed corrected knowledge distillation for TIL. We first describe our implementation details and then demonstrate the performance of our proposed method in overcoming the catastrophic forgetting. Finally, we show its ability to solve the LED problem by comparing the results in the increment-at-once and the increment-in-sequence experiments.

A. Implementation details

We adopt SECONDS$^2$ as our basic 3D object detector. It is an RPN for LiDAR-based 3D object detection. To simplify the

1https://github.com/traveller59/second.pytorch

TABLE I: Comparison among different TIL schemes in our experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>trainable param.</th>
<th>initial lr</th>
<th>training steps per task</th>
<th>anno.</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>fine-tuning</td>
<td>$\phi \cup \varphi \cup \varphi_A$</td>
<td>0.1\gamma</td>
<td>50 epochs</td>
<td>new</td>
<td>$L_{\text{det}}$</td>
</tr>
<tr>
<td>joint training</td>
<td>$\theta$</td>
<td>$\gamma$</td>
<td>50 epochs</td>
<td>all</td>
<td>$L_{\text{det}}$</td>
</tr>
<tr>
<td>evictmas</td>
<td>$\theta$</td>
<td>$\gamma$</td>
<td>50 epochs</td>
<td>new &amp;</td>
<td>$L_{\text{det}} + L_{\text{dist}}$</td>
</tr>
<tr>
<td>incdet</td>
<td>$\theta$</td>
<td>$\gamma$</td>
<td>50 epochs</td>
<td>new &amp;</td>
<td>$L_{\text{det}} + L_{\text{MAP}}$.</td>
</tr>
<tr>
<td>c-kd</td>
<td>$\theta$</td>
<td>$\gamma$</td>
<td>50 epochs</td>
<td>new &amp;</td>
<td>$L_{\text{det}} + L_{\text{MAP}}$.</td>
</tr>
</tbody>
</table>

1For instance, if we discretize the 3D spaces into a 3D grid with the shape $[200 \times 200 \times 8]$, and put ten anchors on each location with representing each anchor with a vector of length seven, then $a \in \mathbb{R}^{200 \times 200 \times 8 \times 10 \times 7}$, where $M = 200 \times 200 \times 8$, $A = 10$, $S = 7$. 
description, we continue to use the notation in Section IV and denote the old task(s) as task-A and the new task(s) as task-B. We train task-A from scratch and consider the following TIL training schemes:

- **fine-tuning**: We freeze the old-task part of the header parameters $\varphi_A \subseteq \varphi$. The parameters of feature extractor $\phi$ and the new-task part $\varphi \setminus \varphi_A$ of the header are trainable with the detection loss $L_{det}$. We set the initial learning rate to 0.1γ to avoid drifting greatly from $\theta^*_A$.
- **joint training**: We merge all the training data of seen tasks and use the annotations of all classes in training. Theoretically, this provides the upper-bound performance for old-task performance in TIL.
- **kd**: It is the implementation of knowledge distillation as ILOD [6]. We additionally consider its two variants: kd (unbiased) with the unbiased sampling strategy and kd (threshold) with the threshold sampling strategy as Faster-ILOD [8]. For more details about the sampling strategies, please refer to our Supplementary Materials.
- **ewc/mas**: It is the implementation of the data-prior-based methods EWC [9] and MAS [11].
- **incdet**: It is the implementation of IncDet [15].
- **c-kd**: It is the implementation of our proposed corrected knowledge distillation. We consider two cases: c-kd (ewc) and c-kd (mas). c-kd (ewc) computes the FIM with equation (8), while c-kd (mas) approximates the FIM with equation (13).

We also list the differences among the TIL schemes in Table I. For more details about our implementation and hyperparameters, please refer to our Supplementary Materials.

**Data**: We use the dataset of KITTI 3D object detection benchmark [31], and consider two more classes “Van” and ”Truck” to construct five tasks: task-1(Car), task-2(Pedestrian), task-3(Cyclist), task-4(Van), and task-5(Truck). The class within the brackets is the task-specific class. Each task is composed of its training set $D_{train}^{(t)} = \{(X_{train,i}^{(t)}, Y_{train,i}^{(t)})\}$ and a testing set $D_{test}^{(t)} = \{X_{test,i}^{(t)}\}$. Every $X_{train,i}^{(t)}$ and $X_{test,i}^{(t)}$ contains at least one instance of the task-specific class. In consequence, all the tasks have different data distributions, annotation distributions and different classes, which coincides with the TIL definition. The statistical information about these tasks is available in our Supplementary Materials. In our experiments, the task-(1..K) represents the merged tasks from task-1 to task-K. We merge tasks by gathering their training datasets and testing datasets. The task-specific class set of task-(1..K) is the union of the tasks from task-1 to task-K, i.e., $C^{(1..K)} = \cup_{i=1..K} C^{(t)}$.

**Evaluation metrics**: We use the 3D average precision (AP) to evaluate the detection results. The intersection-over-union (IoU) thresholds are 0.5 for Car, Van and Truck, and 0.25 for Pedestrian and Cyclist. There are three difficulty levels: easy ($\leq 20$ m), moderate ($\leq 35$ m) and hard ($\leq 50$ m) according to the distances between the object and the ego vehicle as well as the occlusion, as in [31]. We compute the mean 3D average precision (mAP) to compare different cases:

$$mAP^{(t)} = \sum_{c \in C^{(t)}} \frac{N_c^{(t)}}{N^{(t)}} \left[ \frac{1}{3} [A_{c,easy}^{(t)} + A_{c,mod.}^{(t)} + A_{c,hard}^{(t)}] \right],$$

where $C^{(t)}$ denotes the class set of task-$t$, and the mAP is the weighted average of the APs in the three difficulty levels of task-$t$.

**B. Increment at once**

In this experiment, we explore the TIL scenario to increment multiple tasks at once. We first train the 3D detector on task-(1..2) from scratch, an then incrementally train it on task-(3..5). The evaluation results are shown in Table II.

For old tasks, fine-tuning forgets all the old-task knowledge, which shows the manifestation of the catastrophic forgetting. The prior-based methods ewc and mas cannot prevent the detector from forgetting the old-task knowledge in detection tasks, which coincides with the findings in [15]. The knowledge-distillation-based method kd prevents the 3D detector from forgetting and performs better than its unbiased-sampling and threshold-sampling variants. Our corrected knowledge distillation methods c-kd(ewc) and c-kd(mas) perform better or comparable than the original case kd. It shows the effectiveness of knowledge-distillation-based methods in overcoming the the catastrophic forgetting.

For new tasks, all the TIL schemes trained with only new annotations (fine-tuning, kd, ewc/mas, and c-kd) result in much better performance than the cases trained with all or pseudo annotations (joint training, and incdet). We attribute this to the class imbalance of the dataset. In Figure 2, we compare the class histogram of the task-(3..5) and that of the task-(1..5). It demonstrates that the class imbalance is worse in the task-(1..5), which is used in joint training in this experiment. The class-imbalance situation of incdet is similar to the task-(1..5) according to the mechanism of pseudo annotations in IncDet [15].
We can observe the manifestation of LED problem by comparing +B(3)(4)(5) kd in Table III with +B(3..5) kd in Table II: 0.8 mAP ↔ 14.1 mAP in the forget metric. In contrast, c-kd (ewc) and c-kd (mas) perform much better and more consistent: 0.8 mAP ↔ 2.6 mAP as well as 0.0 mAP ↔ 0.1 mAP in the forget metric. It demonstrates that our proposed corrected knowledge distillation method takes effect in solving the LED problem as we expected. We also plot the mAP curves of the old task during the whole TIL process in Figure 6 for a better comparison.

We also conducted the incremental-in-sequence based on the NuScenes dataset [33]. The same conclusion still holds, and please refer to our Supplementary Materials for more detail.

VI. CONCLUSION

In this paper, we attribute the LED problem to the inevitable data distribution differences in TIL. To solve this problem, we propose to correct the original knowledge distillation for TIL by additionally maximizing the posterior probability related to all previously seen tasks. We show its usefulness with an application in 3D object detection. The experimental results demonstrate its effectiveness. Our proposed method reaches the upper-bound performance, which is provided by joint training with all old data, in the TIL experiments based on the KITTI dataset.

As a result, the existing knowledge-distillation-based TIL methods will benefit from the proposed corrected knowledge distillation and prevent parametric models from forgetting knowledge even in the face of a long task sequence.

C. Increment in sequence

We explore an alternative scenario of TIL: incrementing multiple tasks in sequence. To compare with the increment-at-once experiment in Section V-B, we first train the 3D object detector on task-(1..2) as before, and then incrementally train the detector on task-3, task-4, and task-5 in sequence. The evaluation results are shown in Table III.

![Fig. 5: Histograms of the training sets of task-(3..5) and task-(1..5). The y-axis represents the number of data samples. In the x-axis, “Ped.” is for “Pedestrian”. We highlight the task-specific classes of task-(3..5) with red to demonstrate that the class imbalance is worse in task-(1..5).](image1)

![Fig. 6: Increment-in-sequence mAP curves of task-(1..2). The x-axis denotes the time step of evaluating the mAP on task-(1..2). For example, if the number of steps is three, its mAP value is the evaluation result after incrementally training on task-3.](image2)

<table>
<thead>
<tr>
<th>method</th>
<th>old(+)</th>
<th>forget(-)</th>
<th>new(+)</th>
<th>all(+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(1..2)</td>
<td>81.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>+B(3)(4)(5) fine-tuning</td>
<td>0.0</td>
<td>81.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>+B(3)(4)(5) ewc</td>
<td>0.0</td>
<td>81.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td>+B(3)(4)(5) mas</td>
<td>0.0</td>
<td>81.9</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td>+B(3)(4)(5) kd (unbiased)</td>
<td>66.8</td>
<td>15.1</td>
<td>27.9</td>
<td>53.3</td>
</tr>
<tr>
<td>+B(3)(4)(5) kd (threshold)</td>
<td>67.8</td>
<td>14.1</td>
<td>33.5</td>
<td>55.4</td>
</tr>
<tr>
<td>+B(3)(4)(5) incdet</td>
<td>75.3</td>
<td>6.6</td>
<td>25.9</td>
<td>58.7</td>
</tr>
<tr>
<td>+B(3)(4)(5) c-kd (ewc)</td>
<td>79.3</td>
<td>2.6</td>
<td>37.8</td>
<td>65.2</td>
</tr>
<tr>
<td>+B(3)(4)(5) c-kd (mas)</td>
<td>81.8</td>
<td>0.1</td>
<td>39.2</td>
<td>66.8</td>
</tr>
<tr>
<td>+B(3)(4)(5) joint training</td>
<td>83.8</td>
<td>-1.9</td>
<td>29.4</td>
<td>65.3</td>
</tr>
<tr>
<td>A(1..5)</td>
<td>81.5</td>
<td>0.4</td>
<td>28.8</td>
<td>64.8</td>
</tr>
</tbody>
</table>

TABLE III: Evaluation results on testing set of the increment-in-sequence TIL experiment based on the KITTI dataset.

REFERENCES


